PEEK: PARTIAL LABEL LEARNING EMPOWERED BY EXTERNAL KNOWLEDGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Partial Label Learning (PLL) is a challenging weakly supervised learning problem that aims to identify the ground-truth label from a candidate set containing exactly one true label and multiple false positives. Despite steady progress, existing PLL methods predominantly rely on *closed-world* supervision derived from the training data itself, leaving a clear gap in how to effectively incorporate *external knowledge*—especially the open-world semantic priors emerging from recent large-scale foundation models. In this paper, we first establish a theory which shows that external guidance should be assigned a higher weight in the early stage of training and down-weighted in the later, enabling a smooth transition from knowledge-assisted disambiguation to self-driven refinement. Building on this insight, we propose **PEEK**, a computationally efficient framework that *leverages the open-world semantic knowledge* of a frozen VLM (e.g., CLIP). PEEK dynamically fuses the model's confidence with the VLM-derived confidence, either via simple global weighting or instance-adaptive weighting, thereby enabling an adaptive transition from external guidance to internal self-evolution. As a versatile plug-and-play module, PEEK significantly accelerates convergence and improves a wide range of state-of-the-art PLL baselines on standard benchmarks as well as challenging fine-grained datasets, without introducing additional training overhead.

1 INTRODUCTION

The remarkable success of modern deep learning is heavily predicated on the availability of large-scale datasets with precise annotations. However, collecting such high-quality data is often labor-intensive and requires specialized expert knowledge, creating a significant bottleneck for real-world applications. To alleviate the burden of precise annotation, *partial label learning* (PLL) (Cour et al., 2011; Lyu et al., 2019) has emerged as a promising weakly supervised learning paradigm Zhou (2017). In the PLL setting, each training instance is annotated with a set of candidate labels, among which exactly one is the ground-truth label while the others are false positives. Owing to its practical annotation protocol, PLL has been successfully applied to a broad range of real-world scenarios, including web mining Luo & Orabona (2010), automatic image annotation Chen et al. (2017), crowdsourcing, and natural language processing Xiong et al. (2023).

The core challenge of PLL lies in *disambiguation*, i.e., the process of identifying the latent ground-truth label from the noisy candidate set. Early approaches tackled this via averaging-based Hüllermeier & Beringer (2006) or identification-based strategies Jin & Ghahramani (2002), often relying on hand-crafted features. In the deep learning era, significant progress has been made by incorporating self-training Lv et al. (2020), consistency regularization Wu et al. (2022), and contrastive learning Wang et al. (2022) into PLL frameworks.

Despite these advancements, existing deep PLL methods predominantly rely on *closed-world* supervision derived solely from the statistical patterns within the specific training dataset. Recently, vision-language models (VLMs) Chen et al. (2023); Zhang et al. (2023) like CLIP Radford et al. (2021), trained on billion-scale image-text pairs, have demonstrated impressive open-world semantic understanding and robust generalization capabilities. Such open-world priors provide external validation to break the confirmation bias inherent in self-training, especially when internal visual features are insufficient to distinguish the ground truth from false positives. Consequently, how to effectively leverage this external knowledge within the PLL framework to guide the disambiguation process

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

becomes a key problem. A very recent attempt, PartialCLIP He et al. (2025), proposes to directly fine-tune these foundation models. Although it is effective, fine-tuning these large foundation models requires substantial training time and considerable memory consumption (see Appendix D(Table 9) for detailed comparison). Furthermore, in many resource-constrained scenarios Tan & Le (2019); Sandler et al. (2018), the goal is to train a standard lightweight model (e.g., ResNet He et al. (2016)) efficiently, rather than maintaining a massive VLM for inference.

In this paper, we address this gap by introducing external knowledge from VLMs without incurring the high cost of fine-tuning. Specifically, we leverage the external knowledge by fusing the model's internal confidence with the VLM-derived static confidence through a weighting factor. We first propose a theory which demonstrates that external knowledge should be assigned a higher weight during the early stage of training and a lower weight in the later stage. This enables the model to gradually reduce its reliance on external knowledge and shift to internal self-improvement. Building on this theoretical grounding, we propose a plug-in framework **PEEK** (**P**artial label learning **E**mpowered by **E**xternal **K**nowledge). PEEK instantiates this theoretical insight via two weighting strategies. The first is a simple global weighting strategy, which employs a time-dependent decay schedule where the weight of external knowledge decreases as training progresses. The second is an instance-adaptive weighting strategy, assigning a high weight of external knowledge to ambiguous samples while a low weight to confident ones. Extensive experiments show the effectiveness of PEEK.

Our main contributions are summarized as follows:

- We propose to leverage external knowledge to solve the PLL ambiguity without the prohibitive cost of fine-tuning.
- We develop a theory to inform the optimal leverage of external knowledge. Based on this theory, we propose PEEK, which effectively fuses the confidence of the model with the confidence derived from VLM.
- Extensive experiments demonstrate that PEEK outperforms state-of-the-art methods across several benchmarks without additional training overhead.

The remainder of this paper is organized as follows. Section 2 reviews related work, while Sections 3 and 4 present the theoretical analysis and the proposed PEEK framework, respectively. Finally, Section 5 details the experimental results, followed by the conclusion in Section 6.

2 Related Work

2.1 Partial Label Learning

The core challenge of PLL lies in *disambiguation*—identifying the true label from the candidate set. Existing methods can be broadly categorized into average-based strategies Hüllermeier & Beringer (2006); Cour et al. (2011); Zhang & Yu (2015) and identification-based strategies Jin & Ghahramani (2002); Yu & Zhang (2015). Average-based strategies treat all candidate labels equally during training, predicting the label by averaging the model's outputs of all candidates. While simple, these methods are often criticized for failing to distinguish the ground-truth from false positives, leading to suboptimal performance. Identification-based strategies, in contrast, view the ground-truth label as a latent variable. These methods aim to progressively maximize the confidence of the true label while suppressing false positives through iterative refinement, typically employing expectation-maximization Lv et al. (2020) or maximum margin frameworks Yu & Zhang (2015).

In the deep learning era, research has predominantly followed the identification-based paradigm to leverage the powerful representation capabilities of deep neural networks. PRODEN Lv et al. (2020) pioneered this direction by employing a self-training mechanism that progressively identifies true labels using the classifier's own outputs. To provide theoretical guarantees, Feng et al. (2020) proposed the risk-consistent (RC) and classifier-consistent (CC) methods, utilizing importance re-weighting and transition matrices to achieve statistical consistency. Beyond loss functions, advanced regularization techniques have been introduced to further improve disambiguation. Wu et al. (2022) revisited consistency regularization (CRDPLL), aligning model outputs across multiple augmented views to a conformal label distribution inferred via bi-level optimization. Similarly, PiCO Wang et al. (2022) integrates contrastive learning to learn discriminative representations, while PaPi Xia et al.

(2023) guides a prototypical classifier to capture visual similarities. However, these methods typically operate in a *closed-world* manner, relying solely on the intrinsic statistical patterns of the training dataset, which can be limiting. Consequently, they often struggle in high-ambiguity scenarios where internal visual features are insufficient for disambiguation without external semantic guidance.

2.2 Vision-Language Models in Partial Label Learning

Large-scale vision-language models (VLMs), such as CLIP Radford et al. (2021), BLIP Li et al. (2022), and CoCa Yu et al. (2022) have demonstrated remarkable capabilities in learning transferable visual representations by aligning images and text in a shared latent space Radford et al. (2021). A key advantage of these models is their zero-shot inference capability, which allows them to classify images into arbitrary categories defined by natural language prompts without explicit training on the target dataset Zhou et al. (2022); Zhang et al. (2021).

Leveraging VLMs for weakly supervised learning has attracted increasing attention. Recent works have explored using VLMs to correct noisy labels Wei et al. (2024) or generate pseudo-labels for unlabeled data Zhang et al. (2024). In PLL, PartialCLIP He et al. (2025) demonstrates that fine-tuning foundation models using parameter-efficient tuning Prottasha et al. (2025) techniques like Adapters Houlsby et al. (2019) or LoRA Hu et al. (2022) on PLL datasets can significantly outperform small models like ResNets He et al. (2016).

Despite their effectiveness, these fine-tuning paradigms incur substantial training costs. To bridge this gap, in the following sections, we first establish a theory which effectively guides the optimal usage of external knowledge, and based on this theory, we propose the PEEK framework for fusing the model's confidence with the confidence derived from the VLM.

3 Theoretical Analysis

3.1 Problem Formulation

In this subsection, we establish the problem setting and notation for PLL. Let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the d-dimensional feature space and $\mathcal{Y} = \{1, 2, \ldots, C\}$ denote the label space with C classes. In the standard PLL framework, we are given a training dataset $\mathcal{D} = \{(\mathbf{x}_i, S_i)\}_{i=1}^N$ consisting of N instances. For each instance $\mathbf{x}_i \in \mathcal{X}$, $S_i \subseteq \mathcal{Y}$ represents the candidate label set associated with $\mathbf{x}_i$. The fundamental assumption of PLL is that the ground-truth label y_i^* is concealed within the candidate set, *i.e.*, $y_i^* \in S_i$, and any label not in S_i is assumed to be incorrect, *i.e.*, $\forall y \notin S_i, y \neq y_i^*$. Our goal is to learn a multi-class classifier $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}^C$ parameterized by $\boldsymbol{\theta}$ (e.g., a deep neural network). For a given instance $\mathbf{x}$, we apply the softmax function to the model output to obtain the corresponding confidence vector $\mathbf{p} = [p_1, p_2, \ldots, p_C]^\top \in [0, 1]^C$ which satisfies $\sum_{i=1}^C p_i = 1$.

In addition to the trainable model $f_{\boldsymbol{\theta}}$, we leverage a pre-trained vision-language model to aid disambiguation. For each instance $\mathbf{x}_i$, the frozen VLM yields a zero-shot confidence distribution over all C classes. To comply with the PLL assumption that only candidate labels are feasible, we set the confidence of non-candidate labels (i.e., $y \notin S_i$) to zero and renormalize the remaining values over the candidate label set, resulting in the confidence distribution $\mathbf{p}_{\mathrm{T}} \in \mathbb{R}^C$. Crucially, since the parameters of the VLM are frozen, $\mathbf{p}_{\mathrm{T}}$ serves as a static confidence prior. Next, we investigate how to effectively fuse this static confidence prior with the dynamic confidence of the trainable model.

3.2 Theory for Dynamic Weighting

In this subsection, we model the confidence of the VLM and $f_{\boldsymbol{\theta}}$ and present a theoretical derivation of the optimal fusion strategy.

For a sample $\mathbf{x}_i$, we consider its corresponding one-hot ground-truth label $\mathbf{y}^* \in \{0, 1\}^C$. We denote the confidence of the VLM as $\mathbf{p}_{\mathrm{T}} = \mathbf{y}^* + \boldsymbol{\epsilon}_{\mathrm{T}}$, where $\boldsymbol{\epsilon}_{\mathrm{T}}$ is the inherent error in the zero-shot inference of the VLM. As the parameters of the VLM are frozen, its confidence $\mathbf{p}_{\mathrm{T}}$ is fixed and thus the error $\boldsymbol{\epsilon}_{\mathrm{T}}$ is constant. The resulting mean square error (MSE), $\mathcal{E}_{\mathrm{T}} = \mathbb{E}[|\boldsymbol{\epsilon}_{\mathrm{T}}|^2]$, is therefore also a constant. Accordingly, let the prediction confidence of model $f_{\boldsymbol{\theta}}$ at epoch t be $\mathbf{p}_{\mathrm{S}}^{(t)} = \mathbf{y}^* + \boldsymbol{\epsilon}_{\mathrm{S}}^{(t)}$, where $\boldsymbol{\epsilon}_{\mathrm{S}}^{(t)}$ represents the estimation error of the model at epoch t. Assuming the model is in an effective learning

phase, its expected MSE $\mathcal{E}_{\mathrm{S}}(t) = \mathbb{E}[\|\boldsymbol{\epsilon}_{\mathrm{S}}^{(t)}\|^2]$ decreases monotonically with epoch t. We formulate the mixed target from the VLM and $f_{\boldsymbol{\theta}}$ as a linear combination with a coefficient $\alpha_t \in [0, 1]$:

$$\mathbf{p}_{\mathrm{mix}}^{(t)} = (1 - \alpha_t)\mathbf{p}_{\mathrm{S}}^{(t)} + \alpha_t \mathbf{p}_{\mathrm{T}}. \tag{1}$$

Our objective is to find the optimal weight α_t^* that minimizes the total MSE: $\mathcal{L}(\alpha_t) = \mathbb{E}[\|\mathbf{p}_{\mathrm{mix}}^{(t)} - \mathbf{y}^*\|^2]$. Let $\frac{d\mathcal{L}}{d\alpha_t} = 0$, the optimal α_t is:

$$\alpha_t^* = \frac{\mathcal{E}_{\mathrm{S}}(t) - \mathrm{Cov}(t)}{\mathcal{E}_{\mathrm{S}}(t) + \mathcal{E}_{\mathrm{T}} - 2\mathrm{Cov}(t)}, \tag{2}$$

where $\mathrm{Cov}(t) = \mathbb{E}[\langle \boldsymbol{\epsilon}_{\mathrm{S}}^{(t)}, \boldsymbol{\epsilon}_{\mathrm{T}} \rangle]$ represents the covariance between the error of the VLM and the model $f_{\boldsymbol{\theta}}$. The detailed proof is shown in Appendix A.1.

Consequently, the evolution of the optimal weight α_t^* depends on the relative magnitude between the error of the VLM and that of the model. To formalize this relationship, we define the error ratio $\kappa(t)$ as $\kappa(t) = \sqrt{\mathcal{E}_{\mathrm{S}}(t)/\mathcal{E}_{\mathrm{T}}}$. Since the MSE of model $f_{\boldsymbol{\theta}}$ decreases over epochs while that of the VLM is constant, the error ratio $\kappa(t)$ decreases monotonically with t. Under this definition, Eq. (2) can be rewritten as:

$$\alpha^*(\kappa) = \frac{\kappa^2 - \rho\kappa}{\kappa^2 - 2\rho\kappa + 1}, \tag{3}$$

where $\rho = \frac{\mathbb{E}[\boldsymbol{\epsilon}_{\mathrm{T}}^\top \boldsymbol{\epsilon}_{\mathrm{S}}]}{\sqrt{\mathcal{E}_{\mathrm{T}}\mathcal{E}_{\mathrm{S}}}} \in (-1, 1)$ denotes the correlation coefficient between the error vectors $\boldsymbol{\epsilon}_{\mathrm{S}}$ and $\boldsymbol{\epsilon}_{\mathrm{T}}$.

Crucially, $\kappa = 1$ represents the model error equals the VLM error. The optimal weight satisfies $\alpha^* > 0.5$ when $\kappa > 1$ and $\alpha^* < 0.5$ when $\kappa < 1$, with the complete derivation provided in Appendix A.2. Specifically, $\kappa > 1$ indicates that the model is in the early training stages, where the model is relatively weak (i.e., $\mathcal{E}_{\mathrm{S}} > \mathcal{E}_{\mathrm{T}}$). In this regime, the derived optimal weight satisfies $\alpha^* > 0.5$, which implies that the VLM should dominate the supervision, and its confidence should be assigned a higher weight. As the training proceeds to the transition point ($\kappa = 1$), the model's performance matches the VLM, resulting in an equilibrium weight of $\alpha^* = 0.5$. Finally, in the late training stage where $\kappa < 1$, the model captures domain-specific patterns and outperforms the zero-shot VLM (i.e., $\mathcal{E}_{\mathrm{S}} < \mathcal{E}_{\mathrm{T}}$). Meanwhile, the optimal weight shifts to $\alpha^* < 0.5$, which implies that the confidence of the VLM should be given a lower weight to allow the model to rely more on its own predictions. This approach allows the model to benefit from external knowledge in the early training stages and refine its predictions in the later stages, thereby mitigating the bias introduced by the external knowledge.

4 METHOD

Based on the theoretical analysis above, we propose a general framework PEEK that integrates VLM's external knowledge into PLL training. Figure 1 provides an overview of the proposed framework. PEEK provides two implementations of the weighting strategy: a simple global dynamic weighting (GDW) and an instance-specific weighting (ISW).

4.1 GLOBAL DYNAMIC WEIGHTING STRATEGY

To effectively leverage the capability of the VLM while mitigating its potential bias, PEEK first introduces GDW, which is a simple time-dependent decay weight strategy.

Specifically, we denote $w(t)$ as the weight coefficient at epoch t, which is defined by a linear decay schedule:

$$w(t) = \alpha \left(1 - \frac{t}{T_{max}}\right). \tag{4}$$

For an instance $\mathbf{x}_i$ at epoch t, we fuse the confidence of VLM $\mathbf{p}_{\mathrm{T}}(\mathbf{x}_i)$ and the confidence of the model $\mathbf{p}_{\mathrm{S}}^{(t)}(\mathbf{x}_i)$ to obtain the dynamic target distribution $\mathbf{p}_{\mathrm{mix}}^{(t)}(\mathbf{x}_i)$:

$$\mathbf{p}_{\mathrm{mix}}^{(t)}(\mathbf{x}_i) = w(t) \cdot \mathbf{p}_{\mathrm{T}}(\mathbf{x}_i) + (1 - w(t)) \cdot \mathbf{p}_{\mathrm{S}}^{(t)}(\mathbf{x}_i). \tag{5}$$

Here, T_{max} is the total number of training epochs, and $\alpha \in (0, 1]$ is a hyper-parameter controlling the initial weight of VLM's confidence.

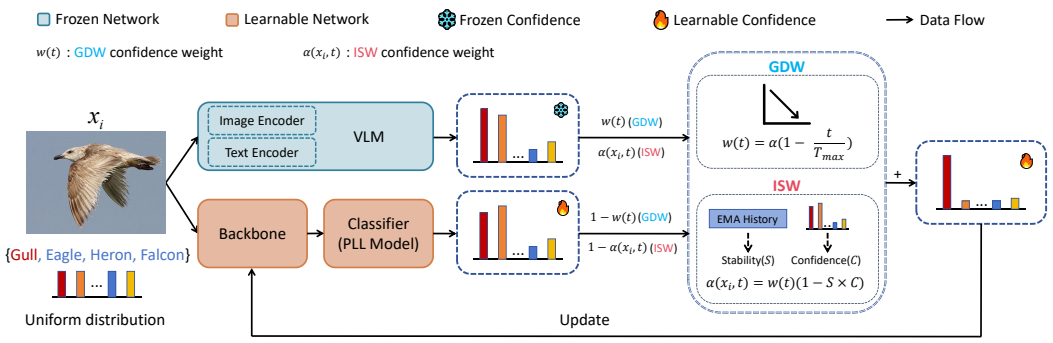


Figure 1: Overview of our proposed PEEK framework. The process consists of two stages: (I) zero-shot inference using a frozen VLM with multi-template prompts to generate static confidence priors; and (II) PLL training where the backbone network is supervised by both the candidate label sets and dynamically weighted guidance from the VLM. In stage (II), we employ two weighting strategies: global dynamic weighting (GDW) and instance-specific weighting (ISW). GDW gradually decays the weight of external knowledge over training epochs, and ISW further modulates the weight based on prediction stability and confidence.

This strategy ensures that external knowledge effectively guides the model during the early training stages, while allowing the model to correct itself and mitigate the negative impact of external knowledge bias in the later stages.

4.2 Instance-Specific Weighting Strategy

While GDW balances the trade-off between the external knowledge and the knowledge from the training data, it applies an identical decay rate to the weight of VLM confidence across all samples. This ignores the fact that the confidence of the VLM can be correct for some samples while misleading for others. When the VLM is incorrect, the weight assigned to its confidence should decay much faster than when it is correct. Therefore, PEEK proposes the ISW strategy.

We reformulate the mixed confidence $\mathbf{p}_{\text{mix}}^{(t)}(\mathbf{x}_i)$ by introducing an instance-specific weighting factor $\alpha(\mathbf{x}_i, t)$:

$$\mathbf{p}_{\text{mix}}^{(t)}(\mathbf{x}_i) = \alpha(\mathbf{x}_i, t) \cdot \mathbf{p}_{\text{T}}(\mathbf{x}_i) + (1 - \alpha(\mathbf{x}_i, t)) \cdot \mathbf{p}_{\text{S}}^{(t)}(\mathbf{x}_i), \tag{6}$$

where $\alpha(\mathbf{x}_i, t)$ is determined by modulating the global decay $w(t)$ with the estimated unreliability of the model:

$$\alpha(\mathbf{x}_i, t) = w(t) \times (1 - \mathcal{S}(\mathbf{x}_i, t) \cdot \mathcal{C}(\mathbf{x}_i, t)) \,. \tag{7}$$

Here, $\mathcal{S}(\mathbf{x}_i, t)$ measures the stability of the model's predictions over time, reflecting its robustness. To define $\mathcal{S}(\mathbf{x}_i, t)$, exponential moving average (EMA Tarvainen & Valpola (2017)) is employed to track the historical prediction $\tilde{\mathbf{p}}_{\text{S}}^{(t)}(\mathbf{x}_i)$:

$$\tilde{\mathbf{p}}_{\text{S}}^{(t)}(\mathbf{x}_i) = \beta \cdot \tilde{\mathbf{p}}_{\text{S}}^{(t-1)}(\mathbf{x}_i) + (1 - \beta) \cdot \mathbf{p}_{\text{S}}^{(t)}(\mathbf{x}_i), \tag{8}$$

where $\beta \in [0, 1)$ is a momentum coefficient. Based on this, the stability score $\mathcal{S}(\mathbf{x}_i, t)$ is then defined as the cosine similarity between the current prediction and the historical trajectory:

$$\mathcal{S}(\mathbf{x}_i, t) = \cos\left(\mathbf{p}_{\text{S}}^{(t)}(\mathbf{x}_i), \tilde{\mathbf{p}}_{\text{S}}^{(t)}(\mathbf{x}_i)\right) \in [0, 1]. \tag{9}$$

Notably, $\mathcal{S}(\mathbf{x}_i, t) \to 1$ implies that the model's prediction has converged and remains stable. $\mathcal{C}(\mathbf{x}_i, t)$ measures the sharpness of the model's prediction distribution. A confident prediction typically exhibits a highly concentrated probability distribution (i.e., low entropy), and therefore $\mathcal{C}(\mathbf{x}_i, t)$ serves as a proxy for the model's certainty. Specifically, $\mathcal{C}(\mathbf{x}_i, t)$ is defined based on the normalized entropy as follows:

$$\mathcal{C}(\mathbf{x}_i, t) = 1 - \frac{H\left(\mathbf{p}_{\text{S}}^{(t)}(\mathbf{x}_i)\right)}{\log |S_i|} \in [0, 1], \tag{10}$$

where $H\left(\mathbf{p}_{\mathrm{S}}^{(t)}(\mathbf{x}_i)\right) = -\sum_j p_j^{(t)}(\mathbf{x}_i) \log p_j^{(t)}(\mathbf{x}_i)$ denotes the entropy. The normalization term $\log |S_i|$ corresponds to the maximum entropy of a uniform distribution over the candidate set. This normalizes the metric to $[0, 1]$, ensuring it remains invariant to the varying candidate set sizes. When the distribution is sharp (i.e., low entropy), $\mathcal{C}(\mathbf{x}_i, t) \to 1$, and when the distribution is uniform (i.e., high entropy), $\mathcal{C}(\mathbf{x}_i, t) \to 0$.

This formulation (Eq. 7) implies a strict reliability condition: the influence of VLM is significantly suppressed ($\alpha(\mathbf{x}_i, t) \to 0$) only when the model is both *stable* ($\mathcal{S}(\mathbf{x}_i, t) \to 1$) and *confident* ($\mathcal{C}(\mathbf{x}_i, t) \to 1$). In other cases where the model exhibits oscillation or uncertainty, the system defaults to a higher weight, thereby retaining VLM's knowledge to prevent label noise overfitting.

5 Experiments

In this section, we systematically evaluate our proposed PEEK framework across multiple benchmarks to demonstrate its effectiveness as a plug-and-play module integrated into existing baselines.

5.1 Experiment Setup

5.1.1 Datasets.

We adopt the widely used benchmark image datasets CIFAR-10 and CIFAR-100 Krizhevsky & Hinton (2009), and three challenging fine-grained image datasets: CUB-200-2011 (CUB-200) Wah et al. (2011), Stanford Cars (CARS-196) Krause et al. (2013), and Stanford Dogs (DOGS-120) Khosla et al. (2011). We manually corrupt these datasets into partially labeled versions using two distinct paradigms:

(1) Standard partial label learning (ST-PLL): This paradigm primarily focuses on *uniform noise distributions*. We adopt the standard flip-probability sampling (FPS) strategy Lv et al. (2020). Under this setting, each incorrect class label is independently added to the candidate set with a fixed probability q, which we extend to extreme noise levels (e.g., up to $q = 0.9$ for CIFAR-10 and $q = 0.3$ for CIFAR-100) beyond standard configurations to validate PEEK's robustness. To ensure that $S_i \neq \{y_i^*\}$ (i.e., all instances ultimately have multiple candidate labels), we flip a random incorrect label into the candidate label set in cases where no false label was introduced.

(2) Instance-dependent partial label learning (ID-PLL): This is a more challenging scenario where candidate labels are *instance-specific*. Following the standard generation procedure widely adopted in Xu et al. (2022); Qiao et al. (2023); Zhao et al. (2026), the probability of an incorrect label being flipped into the candidate set S is strictly proportional to its confidence score predicted by a clean pre-trained network, naturally coupling label noise with visual features.

5.1.2 Baselines.

To demonstrate the effectiveness of PEEK as a versatile plug-and-play module, we systematically evaluate it across a diverse set of state-of-the-art baselines. For standard PLL (ST-PLL), we integrate PEEK into **CRDPLL** Wu et al. (2022), **PiCO** Wang et al. (2022), **PaPi** Xia et al. (2023), **CroSel** Tian et al. (2024), **PRODEN** Lv et al. (2020), and **RC** Feng et al. (2020). For instance-dependent PLL (ID-PLL), we evaluate **CAD** Zhao et al. (2026), **POP** Xu et al. (2022), and **IDGP** Qiao et al. (2023). Detailed descriptions of these baselines are provided in Appendix C.2.

5.1.3 Implementation Details.

Our experiments are conducted in a two-stage manner comprising zero-shot inference and PLL model training utilizing the inferred static confidence. In the first stage, we employ the VLM, i.e., the pre-trained CLIP model (ViT-B/16) Radford et al. (2021) to generate static confidence priors. To ensure inference precision and efficiency, the model is configured in fp32 mode, utilizing a dataset-specific multi-template ensemble strategy (Appendix B) for robust text encoding. In the second stage (PLL training), we strictly adhere to the original experimental configurations of the respective baseline methods. For scenarios where a baseline method was not originally evaluated on a specific dataset (e.g., certain fine-grained datasets), we adopt the configurations detailed in Appendix C. We report the

standard top-1 classification accuracy on the test set as the primary evaluation metric. All experiments are implemented in PyTorch Paszke et al. (2019) and executed on NVIDIA RTX 3090 GPUs.

5.2 Experiment Results

5.2.1 Performance under ST-PLL scenario

We evaluate the proposed PEEK framework on two standard benchmarks (CIFAR-10, CIFAR-100) and three fine-grained datasets (CUB-200, CARS-196, DOGS-120) to demonstrate its effectiveness against various baselines, with results presented in Table 1 and Appendix D (Table 7), respectively.

It is obvious that PEEK consistently boosts performance across all benchmarks. For example, on CIFAR-100, CRDPLL+GDW demonstrates robust performance across varying ambiguity levels, achieving accuracies of **83.42%**, **83.69%**, **83.62%**, and **83.53%** for $q \in \{0.05, 0.1, 0.2, 0.3\}$, respectively. Crucially, our results confirm that the learned model eventually surpasses the zero-shot capability of the VLM: for instance, on CARS-196, CRDPLL+ISW reaches 92.35%, far exceeding the CLIP zero-shot baseline of 63.03%. PEEK also demonstrates exceptional robustness where standard methods collapse. Under extreme ambiguity (CIFAR-10, $q = 0.9$), RC reaches 27.57%, whereas RC+GDW and RC+ISW reach 87.29% and 86.98% respectively. This robustness extends to cold-start challenges on CUB-200 ($q = 0.03$), where PRODEN struggles with an accuracy of 35.47%, while PEEK achieves a significantly higher 61.81%. These results confirm that PEEK effectively guides models with external knowledge, ensuring stable convergence even when intrinsic supervision signals are rendered unreliable by heavy label noise.

5.2.2 Performance under ID-PLL scenario

To address the limitations of uniform noise assumptions, we evaluate PEEK on the more realistic and challenging instance-dependent partial label learning scenario.

As presented in Table 2, PEEK consistently delivers significant performance gains. For instance, with the CAD baseline on CIFAR-10, our +ISW strategy improves the accuracy from 93.57% to 96.06%. Notably, on the more complex CIFAR-100 dataset, PEEK helps POP achieve a 2.3% absolute accuracy gain. Furthermore, the consistent superiority of ISW over GDW across these baselines underscores the necessity of instance-level adaptive weighting when tackling instance-dependent label noise. These results demonstrate that the external knowledge from VLMs effectively accelerates label disambiguation even when the noise is instance-specific and highly ambiguous.

5.3 Ablation Study

5.3.1 Dynamic Weight Strategy vs. Static Weight Strategy

To validate the theoretical necessity of decaying weighting schedule, we compare the proposed dynamic strategy with a static weighting baseline. As shown in Table 3, the dynamic approach consistently outperforms the static strategy, particularly in challenging scenarios. This occurs because a fixed weight fails to accommodate the model's evolving needs: it either relies too heavily on external knowledge in the late stages, inhibiting domain-specific refinement, or provides insufficient guidance in the early stages. To examine the choice of decay function, Appendix D (Table 8) compares linear, cosine, and exponential schedules. Linear and cosine decay perform similarly, while exponential decay is less robust. We therefore use the simple linear schedule as the default.

5.3.2 The Importance of Stability and Confidence in ISW

The stability component in ISW measures the temporal consistency of the model's predictions, protecting against stochastic oscillations. Removing this component (w/o Stability), as detailed in Table 3, leads to a consistent performance decline across most settings, especially on the more challenging CIFAR-100 dataset. For example, at $q = 0.3$ on CIFAR-100, the accuracy drops by 40.11%. This indicates that historical consistency helps smoothen the transition from external guidance to self-training, preventing the model from prematurely fitting to noisy labels. The confidence metric acts as a safety valve, retaining external guidance for uncertain hard samples. Table 3 confirms its vital role in robustness. Removing confidence adjustment (w/o Confidence) also leads to noticeable

Table 1: Accuracy comparison of different strategies applied as plug-ins on CIFAR-10 and CIFAR-100 under various ambiguity levels q. Strategies: +GDW denotes the global dynamic weighting strategy, and +ISW denotes the instance-specific weighting strategy. Results are reported as mean ± std (%), and the best results are highlighted in bold.

Method	CIFAR-10				CIFAR-100			
	q=0.3	q=0.5	q=0.7	q=0.9	q=0.05	q=0.1	q=0.2	q=0.3
CRDPLL	97.28±0.02	97.05±0.05	95.77±0.08	82.64±0.06	82.71±0.10	82.24±0.07	80.97±0.29	64.79±0.13
CRDPLL+GDW	**97.50±0.16**	**97.37±0.10**	**97.04±0.13**	**96.73±0.10**	**83.42±0.11**	**83.69±0.19**	**83.62±0.13**	**83.53±0.29**
CRDPLL+ISW	97.48±0.08	97.34±0.11	97.00±0.09	96.49±0.15	83.32±0.05	83.51±0.20	83.48±0.32	83.24±0.26
PiCO	94.18±0.12	93.58±0.06	92.03±0.07	30.92±0.13	72.74±0.30	59.98±0.13	24.09±0.21	14.21±0.14
PiCO+GDW	93.26±0.14	93.00±0.12	93.97±0.77	**93.89±0.05**	**76.13±0.12**	**75.83±0.32**	**75.09±0.33**	**74.46±0.17**
PiCO+ISW	**94.62±0.09**	**94.38±0.07**	**94.14±0.10**	93.39±0.12	74.32±0.22	74.27±0.27	73.28±0.47	72.19±0.41
PaPi	97.26±0.08	96.90±0.09	96.58±0.07	41.83±0.06	81.60±0.18	81.65±0.27	79.49±0.22	18.60±0.14
PaPi+GDW	97.46±0.04	97.24±0.08	97.00±0.21	**96.52±0.12**	**82.48±0.33**	**82.36±0.23**	**82.01±0.09**	81.52±0.27
PaPi+ISW	**97.48±0.08**	**97.38±0.13**	**97.12±0.08**	96.43±0.18	82.05±0.27	82.12±0.41	81.89±0.12	**81.66±0.23**
CroSel	96.42±0.03	95.75±0.05	94.20±0.24	78.86±4.12	83.81±0.08	83.59±0.28	83.59±0.28	56.60±0.24
CroSel+GDW	**96.62±0.05**	**96.65±0.11**	**96.36±0.12**	**96.04±0.16**	**84.47±0.09**	84.33±0.11	**83.74±0.20**	82.76±0.72
CroSel+ISW	96.56±0.06	96.41±0.13	96.29±0.02	96.03±0.11	84.18±0.03	**84.35±0.04**	83.67±0.06	**83.31±0.08**
PRODEN	85.14±0.05	83.83±0.08	81.09±0.13	52.88±0.15	54.82±0.02	52.74±0.04	26.48±0.07	12.32±0.08
PRODEN+GDW	85.85±0.02	86.76±0.12	**87.16±0.07**	**87.24±0.11**	**56.84±0.09**	56.23±0.13	56.84±0.15	**57.71±0.14**
PRODEN+ISW	**86.14±0.04**	**86.81±0.07**	87.08±0.11	87.20±0.05	56.09±0.06	**56.68±0.12**	**57.50±0.14**	57.46±0.21
RC	85.19±0.34	82.34±0.18	73.95±0.12	27.57±0.26	59.17±0.11	54.19±0.09	25.28±0.17	12.66±0.27
RC+GDW	**87.02±0.07**	**86.80±0.05**	**87.03±0.07**	**87.29±0.08**	**60.58±0.10**	**61.01±0.07**	**59.40±0.06**	58.71±0.13
RC+ISW	86.41±0.17	86.31±0.14	86.41±0.11	86.98±0.09	60.24±0.21	59.49±0.15	58.97±0.09	**59.41±0.17**
CLIP: zero-shot	89.76±0.21				64.26±0.17			

Table 2: Accuracy comparison on CIFAR-10 and CIFAR-100 under instance-dependent partial label noise. Results are reported as mean ± std (%), and the best results are highlighted in bold.

Method	CIFAR-10			CIFAR-100		
	Baseline	+GDW	+ISW	Baseline	+GDW	+ISW
CAD	93.57±0.24	95.70±0.13	**96.06±0.10**	72.03±1.88	73.69±0.69	**73.87±0.41**
POP	94.30±0.10	94.99±0.16	**95.39±0.16**	73.03±0.17	74.66±0.28	**75.33±0.27**
IDGP	88.04±0.25	91.21±0.09	**91.22±0.19**	62.68±0.45	66.00±0.34	**66.07±0.41**

performance degradation in high-noise environments: on CIFAR-100 ($q = 0.3$), accuracy drops by 0.39%. This demonstrates that without confidence-awareness, the global decay schedule discards VLM's supervision too early for difficult samples.

5.3.3 Convergence Analysis

Unlike standard PLL methods that rely solely on closed-world supervision and often suffer from a cold-start dilemma resulting in slow convergence, PEEK utilizes external knowledge to bootstrap the learning process. To verify the claim that PEEK significantly accelerates convergence, we conduct a comparative convergence analysis. As visualized in Appendix D(Figure 2), standard PLL methods suffer from a cold-start phenomenon. This is evident in the test accuracy curves of PiCO (Figure 2a, 2b) and CRDPLL (Figure 2c, 2d) baselines. In contrast, PEEK decisively resolves this cold-start dilemma by leveraging static VLM priors. This results in significantly faster convergence and consistently higher accuracy from the early training stages across all tested baselines, enabling the model to quickly establish robust representations before transitioning to self-evolution.

5.3.4 Parameter Sensitivity Analysis

We also investigate the sensitivity of our method to two key hyper-parameters: the momentum coefficient β in ISW and the initial α value in GDW. As shown in Appendix D(Figure 2(e)–(h)), our

Table 3: Ablation study of the PEEK framework combined with PiCO. We compare the full method against variants using fixed weights (Static), removing the stability component (w/o Stability), and removing the confidence component (w/o Confidence). Best results are highlighted in bold. All values are reported as mean ± std (%).

CIFAR-10					CIFAR-100				
q	Full	Static	w/o Stab	w/o Conf	q	Full	Static	w/o Stab	w/o Conf
0.3	**94.62±0.09**	93.53±0.23	94.13±0.12	94.52±0.06	0.05	**74.32±0.02**	71.22± 0.07	72.89±0.31	74.20±0.24
0.5	**94.38±0.07**	93.35±0.14	93.79±0.20	94.21±0.07	0.1	**74.27±0.27**	70.97±0.39	71.75±0.45	73.96±0.33
0.7	**94.14±0.10**	92.96±0.04	92.34±0.25	94.06±0.01	0.2	**73.28±0.47**	70.19±0.04	52.88±2.21	72.84±0.28
0.9	**93.39±0.12**	92.55±0.28	62.25±5.78	93.29±0.05	0.3	**72.19±0.41**	69.15±0.49	32.08±1.16	71.80±0.51

method is robust to variations in both parameters. For β, performance is maintained across a wide range ($\beta \in [0.30, 0.99]$) on both CIFAR-10 and CIFAR-100, with optimal performance typically achieved within $[0.90, 0.95]$. This aligns with the EMA intuition that a proper momentum effectively balances noise smoothing with responsiveness. Similarly, our method exhibits robustness to the initial guidance weight α, maintaining stable accuracy across varying levels of external supervision intensity. This resilience indicates that PEEK adapts to different initial guidance intensities, preventing final model performance from being overly sensitive to the specific starting configuration.

5.3.5 Robustness to the Quality of External Knowledge

A potential concern is whether PEEK remains effective when high-quality external knowledge or strong pre-trained VLMs are unavailable. To investigate this, we (1) utilize various CLIP backbones on CIFAR-100 (Table 4), and (2) evaluate a challenging FGVC scenario where both the VLM prior and the PLL baseline perform poorly (Table 5). As shown in Table 4, stronger VLMs provide more accurate priors and yield better PLL performance. More importantly, PEEK demonstrates remarkable robustness even when the prior is inaccurate: on FGVC, the frozen CLIP achieves only 25.14% zero-shot accuracy and CRDPLL baseline collapses to 21.87%, yet PEEK boosts performance to 62.53%, showing that sparse external signals can also effectively bootstrap label disambiguation.

Table 4: Ablation study on different CLIP backbones of the PiCO method with our PEEK(ISW) framework on CIFAR-100. Other columns show the best test accuracy (%) under different partial rates (q). "Def." denotes the default configuration using the ViT-B/16 backbone. Best results are highlighted in bold.

Backbone	**Zero-shot**	**PiCO+PEEK** (q)			
		0.05	**0.1**	**0.2**	**0.3**
RN50	31.72	74.76	74.42	72.58	69.78
ViT-B/32	59.17	75.79	75.27	74.78	73.86
ViT-B/16 (Def.)	64.26	74.32	74.27	73.28	72.19
ViT-L/14	**73.95**	**75.52**	**75.77**	**75.30**	**75.42**

Table 5: Robustness evaluation of PEEK using the CRDPLL baseline under low-quality external knowledge on the FGVC dataset. The accuracy results (%) under different partial rates (q) are reported.

Method	**Partial Rate** q			
	0.05	**0.1**	**0.2**	**0.3**
Baseline	21.87	7.44	4.23	2.52
+ GDW	**62.53**	**59.44**	**16.41**	7.74
+ ISW	57.10	15.96	16.56	**10.77**
CLIP: zero-shot	25.14			

6 Conclusion

In this paper, we presented PEEK, a computationally efficient plug-in addressing the PLL cold-start dilemma by leveraging external knowledge. PEEK effectively fuses the model's confidence with the confidence derived from the VLM. Theoretically grounded in a decaying weighting schedule, PEEK proposes GDW and ISW mechanisms, enabling an adaptive transition from external guidance to internal self-evolution. Experiments demonstrate that PEEK accelerates convergence and boosts SOTA performance without additional training overhead, particularly under high ambiguity.

AI USE STATEMENT

In this work, we used generative AI tools for none of the tasks with required disclosure. We have not used generative AI tools for the development of ideas or methods, theoretical claims or proofs, experiments, code, data analysis, or results, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for limited language polishing and improving the clarity of expression. We have reviewed all AI-assisted work. Specifically, the AI-assisted language edits were checked by the authors to ensure that they preserved the original scientific content, claims, and meaning and did not introduce inaccurate or unsupported statements. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This paper presents work whose goal is to advance the field of Machine Learning, specifically in partial label learning (PLL). While our primary goal is to improve algorithmic efficiency and robustness, we acknowledge potential societal consequences.

Our PEEK framework aligns with green AI principles. By utilizing frozen VLMs instead of fine-tuning them, we significantly reduce the computational resources and carbon footprint associated with model training compared to fully supervised or fine-tuning approaches.

We utilize foundation models (e.g., CLIP) which are known to inherit biases from internet-scale training data. There is a risk that such biases could be transferred to downstream tasks. However, our proposed dynamic weighting mechanism is designed to shift the model's reliance from external priors to internal domain-specific data progressively, which may help mitigate the amplification of these inherent biases. We encourage practitioners to audit static priors when applying this method to sensitive domains.

REPRODUCIBILITY STATEMENT

The theoretical assumptions and derivations are provided in Appendix A. Dataset-specific prompts, experimental settings, and additional analyses are described in the remaining appendix sections.

REFERENCES

Ching-Hui Chen, VishalM. Patel, and Rama Chellappa. Learning from ambiguously labeled face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Fei-Long Chen, Du-Zhen Zhang, Ming-Lun Han, Xiu-Yi Chen, Jing Shi, Shuang Xu, and Bo Xu. Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20(1):38–56, 2023.

Timothee Cour, Ben Sapp, and Ben Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12(42):1501–1536, 2011.

Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. In *Advances in Neural Information Processing Systems 33*, pp. 10948–10960, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Kuang He, Wei Tang, Tong Wei, and Min-Ling Zhang. Tuning the right foundation models is what you need for partial label learning. *arXiv preprint arXiv:2506.05027*, 2025.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Eyke Hüllermeier and Jürgen Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–439, 2006.

Rong Jin and Zoubin Ghahramani. Learning with multiple labels. In *Advances in Neural Information Processing Systems*, volume 15, pp. 897–904, 2002.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, 2011.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, 2013.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto, 2009.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, 2022.

Jiancheng Luo and Francesco Orabona. Learning from candidate labeling sets. *Neural Information Processing Systems,Neural Information Processing Systems*, 2010.

Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6500–6510. PMLR, 2020.

Gengyu Lyu, Songhe Feng, Tao Wang, Congyan Lang, and Yidong Li. Gm-pll: Graph matching based partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, 33:521–535, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. 2019.

Nusrat Jahan Prottasha, Upama Roy Chowdhury, Shetu Mohanto, Tasfia Nuzhat, Abdullah As Sami, Md Shamol Ali, Md Shohanur Islam Sobuj, Hafijur Raman, Md Kowsher, and Ozlem Ozmen Garibay. Peft a2z: Parameter-efficient fine-tuning survey for large language and vision models. *arXiv preprint arXiv:2504.14117*, 2025.

Congyu Qiao, Ning Xu, and Xin Geng. Decompositional generation process for instance-dependent partial label learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114. PMLR, 2019.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Shiyu Tian, Hongxin Wei, Yiqun Wang, and Lei Feng. Crosel: Cross selection of confident pseudo labels for partial-label learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19479–19488, 2024.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. Contrastive label disambiguation for partial label learning. *Proceedings of the International Conference on Learning Representations*, 2022.

Wei Wang, Dong-Dong Wu, Jindong Wang, Gang Niu, Min-Ling Zhang, and Masashi Sugiyama. Realistic evaluation of deep partial-label learning algorithms. In *International Conference on Learning Representations*, volume 2025, pp. 86077–86101, 2025.

Tong Wei, Hao-Tian Li, Chun-Shu Li, Jiang-Xin Shi, Yu-Feng Li, and Min-Ling Zhang. Vision-language models are strong noisy label detectors. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Dong-Dong Wu, Deng-Bao Wang, and Min-Ling Zhang. Revisiting consistency regularization for deep partial label learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 24212–24225. PMLR, 2022.

Shiyu Xia, Jiaqi Lv, Ning Xu, Gang Niu, and Xin Geng. Towards effective visual representations for partial-label learning. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15589–15598, 2023.

Limao Xiong, Jie Zhou, Qunxi Zhu, Xiao Wang, Yuanbin Wu, Qi Zhang, Tao Gui, Xuanjing Huang, Jin Ma, and Ying Shan. A confidence-based partial label learning model for crowd-annotated named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 1375–1386, Toronto, Canada, 2023. Association for Computational Linguistics.

Ning Xu, Jiaqi Lv, Biao Liu, Congyu Qiao, and Xin Geng. Progressive purification for instance-dependent partial label learning. 2022.

F.Richard Yu and Min-Ling Zhang. Maximum margin partial label learning. *Asian Conference on Machine Learning,Asian Conference on Machine Learning*, 2015.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022.

Jiahan Zhang, Qi Wei, Feng Liu, and Lei Feng. Candidate pseudolabel learning: Enhancing vision-language models by prompt tuning with unlabeled data. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 60004–60020. PMLR, 2024.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46:5625–5644, 2023.

Min-Ling Zhang and F.Richard Yu. Solving the partial label learning problem: an instance-based approach. *International Conference on Artificial Intelligence*, 2015.

Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Jiao Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *ArXiv*, abs/2111.03930, 2021.

Rui Zhao, Bin Shi, Kai Sun, and Bo Dong. Mitigating instance entanglement in instance-dependent partial label learning. 2026.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1): 44–53, 2017.

A Theoretical Analysis Details

In this appendix, we provide the detailed proofs and analyses supporting the optimal weighting strategy presented in Section 3. We model the frozen VLM output as a static confidence prior and the trainable model $f_{\boldsymbol{\theta}}$ as a dynamic predictor, aiming to minimize the MSE of their weighted combination.

A.1 Derivation of Optimal Dynamic Weighting

Problem Formulation. Let the one-hot ground-truth label be $\mathbf{y}^* \in \{0,1\}^C$. We characterize the confidence outputs of the two components as follows:

- VLM (Static): $\mathbf{p}_{\mathrm{T}} = \mathbf{y}^* + \boldsymbol{\epsilon}_{\mathrm{T}}$, where $\boldsymbol{\epsilon}_{\mathrm{T}}$ is the static error vector. The constant MSE is denoted as $\mathcal{E}_{\mathrm{T}} = \mathbb{E}[\|\boldsymbol{\epsilon}_{\mathrm{T}}\|^2]$.
- Model $f_{\boldsymbol{\theta}}$ (Dynamic): $\mathbf{p}_{\mathrm{S}}^{(t)} = \mathbf{y}^* + \boldsymbol{\epsilon}_{\mathrm{S}}^{(t)}$, where $\boldsymbol{\epsilon}_{\mathrm{S}}^{(t)}$ is the time-varying error vector at epoch t. The dynamic MSE is denoted as $\mathcal{E}_{\mathrm{S}}(t) = \mathbb{E}[\|\boldsymbol{\epsilon}_{\mathrm{S}}^{(t)}\|^2]$.

The mixed supervision target at epoch t is defined as $\mathbf{p}_{\mathrm{mix}}^{(t)} = (1-\alpha_t)\mathbf{p}_{\mathrm{S}}^{(t)} + \alpha_t \mathbf{p}_{\mathrm{T}}$. Our objective is to find the optimal weight α_t^* that minimizes the total MSE of the mixture:

$$\mathcal{L}(\alpha_t) = \mathbb{E}[\|\mathbf{p}_{\mathrm{mix}}^{(t)} - \mathbf{y}^*\|^2]. \tag{11}$$

MSE Expansion and Solution. First, we substitute the definitions of $\mathbf{p}_{\mathrm{T}}$ and $\mathbf{p}_{\mathrm{S}}^{(t)}$ into the error term:

$$\mathbf{p}_{\mathrm{mix}}^{(t)} - \mathbf{y}^* = (1-\alpha_t)(\mathbf{y}^* + \boldsymbol{\epsilon}_{\mathrm{S}}^{(t)}) + \alpha_t(\mathbf{y}^* + \boldsymbol{\epsilon}_{\mathrm{T}}) - \mathbf{y}^* = (1-\alpha_t)\boldsymbol{\epsilon}_{\mathrm{S}}^{(t)} + \alpha_t \boldsymbol{\epsilon}_{\mathrm{T}}. \tag{12}$$

Next, we expand the squared norm in the objective function, incorporating the covariance term $\mathrm{Cov}(t) = \mathbb{E}[\langle \boldsymbol{\epsilon}_{\mathrm{S}}^{(t)}, \boldsymbol{\epsilon}_{\mathrm{T}} \rangle]$:

$$\begin{aligned}
\mathcal{L}(\alpha_t) &= \mathbb{E}[\|(1-\alpha_t)\boldsymbol{\epsilon}_{\mathrm{S}}^{(t)} + \alpha_t \boldsymbol{\epsilon}_{\mathrm{T}}\|^2] \\
&= (1-\alpha_t)^2 \mathbb{E}[\|\boldsymbol{\epsilon}_{\mathrm{S}}^{(t)}\|^2] + \alpha_t^2 \mathbb{E}[\|\boldsymbol{\epsilon}_{\mathrm{T}}\|^2] + 2\alpha_t(1-\alpha_t)\mathbb{E}[\langle \boldsymbol{\epsilon}_{\mathrm{S}}^{(t)}, \boldsymbol{\epsilon}_{\mathrm{T}} \rangle] \\
&= (1-\alpha_t)^2 \mathcal{E}_{\mathrm{S}}(t) + \alpha_t^2 \mathcal{E}_{\mathrm{T}} + 2\alpha_t(1-\alpha_t)\mathrm{Cov}(t).
\end{aligned} \tag{13}$$

To minimize $\mathcal{L}(\alpha_t)$, we compute its derivative with respect to α_t and set it to zero:

$$\frac{d\mathcal{L}}{d\alpha_t} = -2(1-\alpha_t)\mathcal{E}_{\mathrm{S}}(t) + 2\alpha_t \mathcal{E}_{\mathrm{T}} + 2(1-2\alpha_t)\mathrm{Cov}(t) = 0. \tag{14}$$

Rearranging the terms to solve for α_t, we obtain the optimal weight formulation:

$$\alpha_t^* = \frac{\mathcal{E}_{\mathrm{S}}(t) - \mathrm{Cov}(t)}{\mathcal{E}_{\mathrm{S}}(t) + \mathcal{E}_{\mathrm{T}} - 2\mathrm{Cov}(t)}. \tag{15}$$

This completes the proof for Eq. (2) in the main text.

A.2 Analysis of Weight Trajectory

To strictly analyze the evolution of α_t^* during the training process, we utilize the error ratio $\kappa(t) = \sqrt{\mathcal{E}_{\mathrm{S}}(t)/\mathcal{E}_{\mathrm{T}}}$ and the correlation coefficient $\rho = \mathrm{Cov}(t)/\sqrt{\mathcal{E}_{\mathrm{S}}(t)\mathcal{E}_{\mathrm{T}}}$.

By substituting $\mathcal{E}_{\mathrm{S}}(t) = \kappa^2 \mathcal{E}_{\mathrm{T}}$ and $\mathrm{Cov}(t) = \rho\kappa\mathcal{E}_{\mathrm{T}}$ into the optimal weight equation derived above, we rewrite α_t^* as a function of κ:

$$\alpha^*(\kappa) = \frac{\kappa^2\mathcal{E}_{\mathrm{T}} - \rho\kappa\mathcal{E}_{\mathrm{T}}}{\kappa^2\mathcal{E}_{\mathrm{T}} + \mathcal{E}_{\mathrm{T}} - 2\rho\kappa\mathcal{E}_{\mathrm{T}}} = \frac{\kappa^2 - \rho\kappa}{\kappa^2 - 2\rho\kappa + 1}. \tag{16}$$

To determine the relative importance of the VLM prior versus the model prediction, we compare α^* against the neutral threshold of 0.5 by analyzing the sign of their difference:

$$\begin{aligned}
\alpha^*(\kappa) - 0.5 &= \frac{\kappa^2 - \rho\kappa}{\kappa^2 - 2\rho\kappa + 1} - \frac{0.5(\kappa^2 - 2\rho\kappa + 1)}{\kappa^2 - 2\rho\kappa + 1} \\
&= \frac{0.5\kappa^2 - 0.5}{\kappa^2 - 2\rho\kappa + 1} \\
&= \frac{0.5(\kappa^2 - 1)}{(\kappa - \rho)^2 + (1 - \rho^2)}.
\end{aligned} \tag{17}$$

Since the correlation coefficient satisfies $|\rho| < 1$, the term $(1 - \rho^2)$ is strictly positive, ensuring the denominator is always positive. Consequently, the sign of $\alpha^*(\kappa) - 0.5$ is determined solely by the numerator term $(\kappa^2 - 1)$.

Given that $\mathcal{E}_{\mathrm{S}}(t)$ decreases via optimization while $\mathcal{E}_{\mathrm{T}}$ remains constant, $\kappa(t)$ monotonically decreases over time. This divides the training process into three distinct phases:

1. Early Stage ($\kappa > 1$): In the initial phase, the trainable model is under-fitting and relatively weak compared to the pre-trained VLM (i.e., $\mathcal{E}_{\mathrm{S}} > \mathcal{E}_{\mathrm{T}}$). This implies $\kappa^2 - 1 > 0$, yielding $\alpha^* > 0.5$. This theoretical result justifies assigning a dominant weight to the static VLM prior early in training to mitigate noise.
2. Transition Point ($\kappa = 1$): When the performance of the model matches that of the VLM (i.e., $\mathcal{E}_{\mathrm{S}} = \mathcal{E}_{\mathrm{T}}$), we have $\kappa^2 - 1 = 0$, leading to $\alpha^* = 0.5$. At this equilibrium, both sources of supervision contribute equally.
3. Late Stage ($\kappa < 1$): As the model converges and learns domain-specific patterns, it eventually outperforms the zero-shot VLM (i.e., $\mathcal{E}_{\mathrm{S}} < \mathcal{E}_{\mathrm{T}}$). This implies $\kappa^2 - 1 < 0$ and consequently $\alpha^* < 0.5$. This confirms that in the final stage, the influence of the external knowledge should be down-weighted to allow the model to rely more on its own refined predictions.

In conclusion, the optimal weight α^* naturally necessitates a transition from a high value (> 0.5) to a low value (< 0.5). This analysis provides the theoretical grounding for our proposed PEEK framework.

B Dataset-Specific Text Prompts for Word Embedding

To improve the robustness of the zero-shot static confidence priors, we employ a multi-template ensemble strategy tailored to the characteristics of each dataset. By averaging the text features extracted from multiple descriptive prompts, we cover diverse linguistic expressions and reduce the dependency on a single template. The specific templates used for each dataset in our experiments are listed below.

CIFAR-10 and CIFAR-100

For the generic object recognition tasks in CIFAR-10 and CIFAR-100, we use a single concise template due to the general nature of the classes.

- `"a photo of a {class}."`

CUB-200-2011

For fine-grained bird classification, the templates emphasize the "bird" attribute and include both generic and specific descriptions.

- `"a photo of a {class} bird."`
- `"a picture of a {class} bird."`
- `"an image of a {class} bird."`
- `"{class}"`
- `"a photo of the {class}."`

STANFORD DOGS

Similar to CUB-200, the templates for Stanford Dogs explicitly mention the "dog" category to reduce ambiguity.

- `"a photo of a {class} dog."`
- `"a picture of a {class} dog."`
- `"an image of a {class} dog."`
- `"{class}"`
- `"a photo of the {class}."`

STANFORD CARS

Recognizing the high inter-class similarity in car models, we employ a comprehensive set of 19 templates. These cover various synonyms (automobile, vehicle), viewpoints (side view, front view), and specific car types (sedan, SUV, coupe, etc.) to capture fine-grained visual features.

- `"a photo of a {class} car."`
- `"a picture of a {class} car."`
- `"an image of a {class} car."`
- `"{class}"`
- `"a photo of the {class}."`
- `"a {class} automobile."`
- `"a {class} vehicle."`
- `"a photo of a {class} automobile."`
- `"a side view of a {class}."`
- `"a front view of a {class}."`
- `"a {class} model car."`
- `"a {class} sedan."`
- `"a {class} sports car."`
- `"a {class} luxury car."`
- `"a {class} SUV."`
- `"a {class} truck."`
- `"a {class} convertible."`
- `"a {class} coupe."`
- `"a {class} hatchback."`
- `"a {class} station wagon."`

C EXPERIMENT DETAILS

In this section, we provide a detailed description of the datasets used in our evaluation and the specific implementation details for reproducibility. All experiments are implemented in PyTorch and executed on NVIDIA RTX 3090 GPUs.

C.1 DATASETS

We evaluate our proposed PEEK framework and baselines on two standard benchmarks (CIFAR-10, CIFAR-100) and three challenging fine-grained datasets (CUB-200-2011, Stanford Cars, Stanford Dogs). The details of these datasets are as follows:

- CIFAR-10 consists of 60,000 32×32 color images in 10 classes, with 50,000 training images and 10,000 test images.
- CIFAR-100 consists of 60,000 32×32 color images in 100 classes, containing 500 training images and 100 testing images per class.
- CUB-200-2011 (CUB-200) contains 11,788 images of 200 bird species. We use the standard split with 5,994 training images and 5,794 test images.
- Stanford Cars (CARS-196) contains 16,185 images of 196 car classes, split into 8,144 training images and 8,041 test images.
- Stanford Dogs (DOGS-120) contains 20,580 images of 120 dog breeds. Following common practice, we use 12,000 images for training and 8,580 for testing.

C.2 BASELINES

To demonstrate the effectiveness of the PEEK framework as a general-purpose plug-in, we integrate it into five standard partial label learning algorithms and three instance-dependent partial label learning algorithms.

Standard PLL (ST-PLL) methods:

- **CRDPLL** Wu et al. (2022): Revisits consistency regularization by aligning model outputs across multiple augmented views to a conformal label distribution inferred via bi-level optimization.
- **PiCO** Wang et al. (2022): A contrastive learning framework that leverages learned prototypes to identify the ground-truth label.
- **PaPi** Xia et al. (2023): Utilizes guided prototypical representation learning to facilitate effective label disambiguation during model training.
- **CroSel** Tian et al. (2024): Leverages historical predictions to cross-select confident pseudo-labels across dual models and employs a co-mix consistency regularization to prevent sample waste.
- **PRODEN** Lv et al. (2020): A self-training approach that progressively identifies true labels based on the classifier's predictions.
- **RC** Feng et al. (2020): A theoretically grounded method that achieves risk consistency through an importance re-weighting strategy.

Instance-dependent PLL (ID-PLL) methods:

- **CAD** Zhao et al. (2026): Resolves instance entanglement in ID-PLL by aligning label-specific augmentations and penalizing confusing non-candidate labels.
- **POP** Xu et al. (2022): Iteratively removes incorrect candidate labels to converge to the Bayes optimal classifier.
- **IDGP** Qiao et al. (2023): Models candidate label generation via categorical and Bernoulli distributions, using maximum a posteriori estimation with iteratively refined conjugate priors.

C.3 EXPERIMENTAL SETTINGS

To ensure a fair comparison, we strictly adhere to the original configurations of the baseline methods. For the PLL training stage, we employ the same backbone architectures, optimizers, and learning rate schedules as the official implementations. For fine-grained datasets where official baseline configurations were unavailable, we adopted the specific settings detailed in Table 6.

PEEK Configuration. For our PEEK-integrated versions, we incorporate the generated static confidence from CLIP as external priors to dynamically refine the label disambiguation process, while strictly keeping all other training hyper-parameters (backbone, learning rate, batch size, etc.) identical to the baselines listed in Table 6. This ensures that the performance improvement is solely attributed to the proposed strategy rather than hyperparameter tuning.

Table 6: Detailed experimental configurations for all evaluated baselines. We strictly adopt the default network architectures (backbones) and hyperparameters from their respective original papers. 'Fine-Grained' refers to the CUB-200, CAR-196, and DOGS-120 datasets. Models are trained from scratch unless marked with †, which denotes ImageNet pre-training. For the PEEK framework, the base fusion weight $\alpha = 0.8$ and stability momentum $\beta = 0.9$ are applied uniformly across all methods, highlighting its robustness without method-specific tuning

Method	Dataset	Backbone	Epochs	Batch Size	Initial LR	PEEK Config	
						α	β
CRDPLL	CIFAR-10/100	WideResNet-34-10	200	256	0.1	0.8	0.9
	Fine-Grained	ResNet-34†	200	128	0.01	0.8	0.9
PiCO	CIFAR-10/100	ResNet-18	800	256	0.01	0.8	0.9
	CUB-200	ResNet-18†	300	128	0.01	0.8	0.9
	CAR-196 / DOGS-120	ResNet-18	300	64	0.01	0.8	0.9
PaPi	CIFAR-100	ResNet-18	500	256	0.1	0.8	0.9
	CIFAR-10	ResNet-18	500	256	0.05	0.8	0.9
	Fine-Grained	ResNet-18	500	64	0.05	0.8	0.9
CroSel	CIFAR-10/100	WideResNet-34-10	200	64	0.1	0.8	0.9
PRODEN	CIFAR-10/100	ConvNet	200	256	0.1	0.8	0.9
	Fine-Grained	ResNet-34†	200	64	0.05	0.8	0.9
RC	CIFAR-10	ResNet-32	250	256	0.001	0.8	0.9
	CIFAR-100	ResNet-56	250	256	0.001	0.8	0.9
	Fine-Grained	ResNet-34†	250	128	0.01	0.8	0.9
Instance-Dependent PLL (ID-PLL) Baselines							
CAD	CIFAR-10/100	ResNet-34	800	64	0.01	0.8	0.9
POP	CIFAR-10/100	WideResNet-28-2	500	256	0.05	0.8	0.9
IDGP	CIFAR-10/100	ResNet-32	500	256	0.01	0.8	0.9

C.4 Integration Details of PEEK into Baselines

A core contribution of PEEK is its plug-and-play capability. Rather than altering the inherent loss functions or architectural designs of existing PLL methods, PEEK seamlessly injects the external visual-language prior into the specific confidence-updating phase of each baseline. The detailed integration mechanisms are outlined below:

PRODEN, CRDPLL, RC, and POP: These methods explicitly maintain and update a label confidence matrix during training. For these baselines, PEEK is directly integrated during the confidence update step. We linearly fuse the pre-computed CLIP zero-shot confidence distribution with the model's internally estimated confidence, weighted by our proposed GDW or ISW strategies. The resulting fused confidence is then utilized for subsequent loss calculations and label disambiguation.

PiCO: PiCO employs contrastive learning and maintains a confidence matrix updated via an exponential moving average (EMA) using one-hot pseudo-labels derived from prototype similarities. To integrate PEEK, we intervene immediately before this EMA update. Instead of exclusively relying on the hard one-hot pseudo-labels, we linearly interpolate them with the CLIP zero-shot confidence using our dynamic weighting strategy. This fused, soft target is then fed into the original EMA updater, providing semantic-aware guidance that prevents the model from prematurely collapsing into incorrect prototypes.

PaPi: PaPi focuses on visual similarity-guided prototypes. It computes a normalized pseudo-label distribution by masking the classifier's softmax output with the candidate set, which is subsequently fused with historical confidences via EMA. Similar to PiCO, we inject the CLIP prior prior to the EMA step. We linearly interpolate the model's predicted distribution with the normalized CLIP confidence. This semantically enhanced distribution is then passed to the original EMA function, ensuring that the progressive self-correction is continuously guided by external knowledge.

CroSel: CroSel employs a cross selection strategy alongside a co-mix consistency regularization, where the model's sharpened predictions are masked by the candidate set and normalized to generate

pseudo-targets. We integrate PEEK by directly refining these pseudo-targets with the CLIP zero-shot semantic prior. Specifically, the pre-computed CLIP confidence is first masked within the candidate set and normalized to form a semantic distribution. We then linearly interpolate the model's self-generated pseudo-targets with this CLIP prior using our temporally decaying weight. This intervention ensures that the label selection process is guided by robust external semantic knowledge in the early stages, while smoothly transitioning to the model's internal predictions as the fusion weight decays to zero, effectively mitigating the risk of false-positive selections.

CAD: As an ID-PLL method, CAD focuses on category decoupling and the utilization of non-candidate labels. The original method produces a softmax probability distribution from the encoder for augmented images, which is then decomposed into candidate and non-candidate components via a positive-negative normalization process. These components are normalized and aggregated to form a fused confidence distribution, serving as a soft label to supervise the classification loss and provide weights for contrastive learning. Our PEEK framework refines this fused confidence by introducing the pre-computed CLIP zero-shot prior. Specifically, the CLIP confidence is first masked to the candidate label set and normalized, then linearly interpolated with the baseline's fused confidence using a temporally decaying weight.

IDGP: IDGP models instance-dependent noise utilizing graph-based prototypes. It maintains a continuous confidence array for the main classifier alongside a binary weight array for an independent graph model. We integrate the CLIP prior exclusively into the iterative update phase of the main classifier's confidence array. In each training iteration, rather than relying solely on the classifier's own output, we fuse its prediction with the CLIP semantic distribution using our dynamic weighting strategy. This fusion provides a more robust disambiguation signal to guide the adjustment of the confidence array, especially in the early stages when the classifier's internal predictions are unreliable. Importantly, we strictly confine the CLIP intervention to the main classifier's confidence array and do not alter the graph model's binary weights, thereby introducing external priors while preserving IDGP's intrinsic graph-based discrimination mechanism.

C.5 Architectural Specifications of CLIP Variants

In this work, we utilize four distinct CLIP visual encoder variants: ResNet-50 (RN50), ViT-B/32, ViT-B/16, and ViT-L/14. These models are based on two fundamental architectures: convolutional neural networks (CNN) and vision transformers (ViT). Their specific architectural details are described below.

ResNet-50 (RN50). The RN50 model adheres to the standard ResNet-50 architecture with approximately 77M parameters. It processes images at a resolution of 224×224 and consists of 50 convolutional layers organized into four stages with bottleneck blocks, where the channel width expands from a base of 64 to 2048 in the final stage. Unlike traditional supervised ResNets, the CLIP variant replaces the global average pooling layer with an attention pooling mechanism to aggregate spatial features into a global representation.

ViT-B/32 (Vision Transformer Base, 32×32). This variant adopts the standard Vision Transformer architecture configured with a coarse patch size, containing approximately 86M parameters. The input image (224×224) is divided into non-overlapping patches of size 32×32, resulting in a grid of $7 \times 7 = 49$ patches. When combined with the prepended class token ([CLS]), the effective sequence length is 50. In terms of network depth and width, it utilizes 12 Transformer encoder layers, a hidden dimension of 768, and 12 attention heads.

ViT-B/16 (Vision Transformer Base, 16×16). Sharing the identical depth (12 layers) and width (768 hidden dimension) as the ViT-B/32 variant, ViT-B/16 distinguishes itself by employing a finer patch size of 16×16. This reduction in patch size significantly increases the spatial resolution of the internal representation, resulting in a $14 \times 14 = 196$ patch grid (total sequence length of 197). This configuration allows the model to capture more fine-grained visual details, bringing the total parameter count to approximately 150M.

ViT-L/14 (Vision Transformer Large, 14×14). As a scaled-up "Large" variant, ViT-L/14 is designed for higher capacity with approximately 428M parameters. It processes images with a patch

size of 14×14, yielding a sequence of 257 tokens (256 patches plus one [CLS] token). Architecturally, it is significantly deeper and wider than the Base models, featuring 24 Transformer layers, a hidden dimension of 1024, and 16 attention heads. This substantial increase in scale contributes to its superior representational power at the cost of increased computational complexity.

D Additional Experimental Results

Table 7 reports the detailed results on three fine-grained datasets. Table 8 supplements the dynamic-weight ablation in Section 5.3 by comparing three decay schedules.

Table 7: Accuracy comparison on CUB-200, CARS-196, and DOGS-120 under various ambiguity levels q. +GDW: global dynamic weighting; +ISW: instance-specific weighting. Best accuracies(%) are highlighted in bold.

Method	CUB-200				CARS-196				DOGS-120			
	q=0.03	q=0.05	q=0.1	q=0.15	q=0.03	q=0.05	q=0.1	q=0.15	q=0.05	q=0.1	q=0.2	q=0.3
CRDPLL	60.51	45.58	25.80	15.64	56.00	31.85	17.30	10.42	73.40	69.07	22.81	7.44
CRDPLL+$_{\text{GDW}}$	**76.30**	**76.18**	**74.63**	**73.14**	**92.43**	**92.35**	**91.47**	**90.70**	**74.89**	**75.48**	**74.71**	**73.11**
CRDPLL+$_{\text{ISW}}$	75.20	75.20	72.82	72.25	92.35	92.05	91.26	90.69	74.20	73.48	72.60	70.79
PiCO	65.54	68.00	60.91	44.33	24.22	9.21	3.41	2.11	53.06	26.93	6.76	3.81
PiCO+$_{\text{GDW}}$	73.54	72.68	**72.48**	**71.09**	84.17	**83.35**	**81.62**	**79.31**	**67.80**	**65.20**	**63.17**	**63.19**
PiCO+$_{\text{ISW}}$	**74.53**	**73.94**	70.01	62.15	**85.02**	81.74	66.79	44.63	66.21	62.09	58.25	53.03
PaPi	60.83	10.75	2.78	1.59	50.62	4.24	1.52	1.34	68.66	2.11	1.64	1.56
PaPi+$_{\text{GDW}}$	78.77	75.64	54.17	45.68	92.41	91.52	79.74	74.56	73.51	71.57	64.62	59.82
PaPi+$_{\text{ISW}}$	**79.77**	**79.29**	**77.19**	**73.78**	**92.64**	**92.56**	**92.65**	**92.30**	**74.67**	**73.07**	**70.83**	**69.06**
PRODEN	35.47	22.23	13.24	8.23	22.05	12.57	6.44	4.30	75.98	70.96	54.07	36.70
PRODEN+$_{\text{GDW}}$	61.11	59.37	57.25	55.23	**62.03**	**62.04**	59.17	57.82	76.43	74.09	69.60	68.03
PRODEN+$_{\text{ISW}}$	**61.81**	**60.08**	**59.13**	**56.14**	62.02	61.66	**60.08**	**58.50**	**77.44**	**75.98**	**72.40**	**70.34**
RC	0.52	1.23	1.04	0.88	1.37	1.27	0.86	0.77	1.08	1.33	1.34	0.93
RC+$_{\text{GDW}}$	10.67	8.65	6.83	**10.60**	1.63	1.70	1.68	1.89	7.31	**8.12**	7.20	6.75
RC+$_{\text{ISW}}$	**11.34**	**11.30**	**9.10**	9.23	**2.57**	**2.16**	**1.83**	**2.69**	**7.77**	8.08	**7.66**	**8.55**
CLIP: zero-shot	49.26				63.03				62.54			

Decay-schedule comparison. PEEK uses the linear schedule in Eq. (4). Table 8 shows that cosine decay is close to linear on CRDPLL/CIFAR-10, whereas exponential decay degrades more at high ambiguity. These three-seed results support linear decay as a simple and effective default.

Table 8: CRDPLL+GDW accuracy (%) on CIFAR-10 for three decay schedules (mean±std over three seeds).

Schedule	$q = 0.3$	$q = 0.5$	$q = 0.7$	$q = 0.9$
Linear	97.50±0.16	97.37±0.10	97.04±0.13	96.73±0.10
Cosine	97.41±0.15	97.29±0.09	97.09±0.15	96.77±0.13
Exponential	97.49±0.12	97.21±0.20	96.55±0.08	95.40±0.18

D.1 Computational Cost Comparison with PartialCLIP

We compare PEEK with PartialCLIP He et al. (2025) from complementary memory and runtime perspectives. Table 9 reports CIFAR-100 accuracy and peak training memory under the same ambiguity level. PEEK retains the original CRDPLL backbone and cached priors, using 6.74 GB of peak memory, while PartialCLIP adapts a CLIP ViT-B/16 and uses 15.73 GB. PartialCLIP obtains higher accuracy in this setting, whereas PEEK provides a lighter integration with substantially lower memory use.

Table 9: Accuracy and peak training memory on CIFAR-100 ($q = 0.2$), measured on one RTX 3090 with batch size 64. The methods use different backbones. Test accuracy is mean±std over five seeds.

Method	Backbone (Params)	Test Acc (%)	Peak Memory
CRDPLL	WideResNet (∼46M)	80.97 ± 0.29	6.74 GB
CRDPLL + PartialCLIP	CLIP ViT-B/16 (∼150M total)	**85.51 ± 0.20**	15.73 GB
CRDPLL + PEEK (ISW)	WideResNet (∼46M)	83.48 ± 0.32	**6.74 GB**

Table 10 reports single-seed CRDPLL/CIFAR-10 measurements on a V100. The methods use different backbones, batch sizes, and epoch counts, so per-epoch and total time capture different aspects of cost. PEEK adds little cost within an epoch and uses substantially less peak memory; under the reported training budgets, its 200-epoch protocol takes longer in total than PartialCLIP's 20-epoch protocol. Offline CLIP ViT-B/16 prior generation is a one-time cost: on an RTX 3090 (FP32, batch size 256), it takes 192.12 seconds end to end on CIFAR-10, including model loading, text encoding, train/test image inference, and saving.

Table 10: Measured CIFAR-10 cost. PEEK values average the four ambiguity settings from one instrumented seed; PartialCLIP values average three seeds and the four settings. Hardware and training protocols differ, so totals are descriptive.

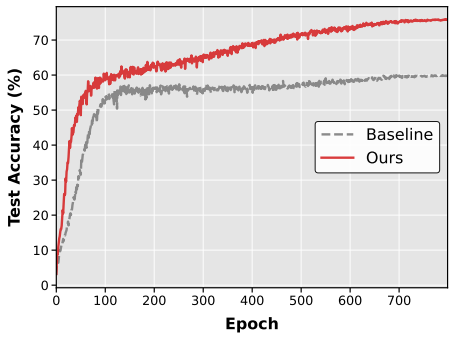
Method	Epochs	Time/epoch (s)	Approx. total train (h)	Peak GPU	Extra state
CRDPLL baseline	200	253.23	14.07	6.31 GB	–
PEEK-GDW	200	264.36	14.69	6.31 GB	cached prior
PEEK-ISW	200	259.07	14.39	6.31 GB	prior + 1.91 MB EMA
PartialCLIP	20	1156.96	6.64	15.68 GB	adapted ViT-B/16

D.2 Results on PLCIFAR10

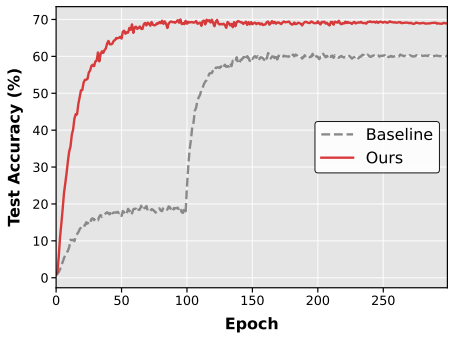
Table 11 evaluates five baselines on PLCIFAR10 Wang et al. (2025), which provides Aggregate and Vaguest human partial annotations for CIFAR-10. Each entry is the best validation accuracy over training, summarized over three seeds. PEEK improves every listed baseline in both protocols, while the relative performance of GDW and ISW remains method dependent.

Table 11: PLCIFAR10 validation accuracy (%, mean±std over three seeds) with human partial annotations.

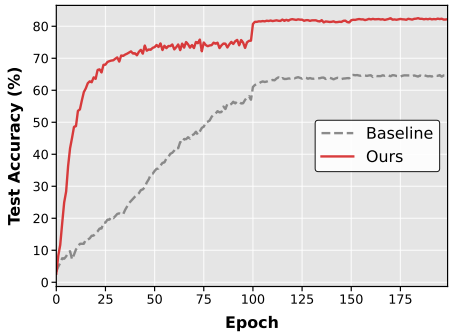
Protocol	Baseline	Original	+GDW	+ISW
Aggregate	PiCO	89.10±0.12	**94.05±0.09**	93.09±0.20
	PaPi	95.11±0.18	96.12±0.14	**96.26±0.14**
	PRODEN	84.50±0.20	**88.63±0.15**	88.53±0.40
	RC	84.53±0.06	**88.50±0.00**	88.23±0.60
	CroSel	92.00±0.14	**95.41±0.07**	95.37±0.11
Vaguest	PiCO	79.67±0.15	**90.84±0.04**	86.18±0.18
	PaPi	84.13±0.33	**89.35±0.16**	89.14±0.17
	PRODEN	72.00±0.52	79.43±0.42	**80.10±0.53**
	RC	71.83±0.67	79.03±0.12	**79.73±0.21**
	CroSel	85.78±0.23	**90.98±0.17**	90.95±0.36



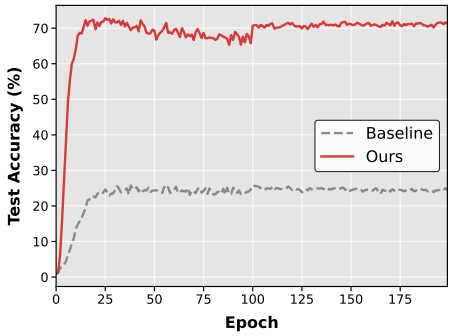
(a) PiCO, CIFAR-100



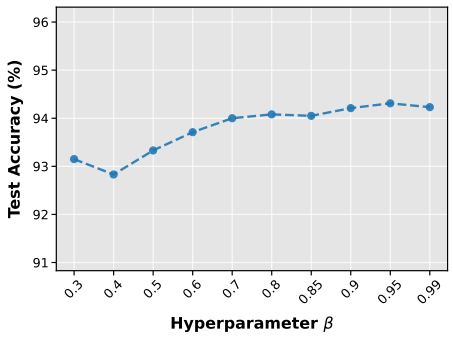
(b) PiCO, CUB-200



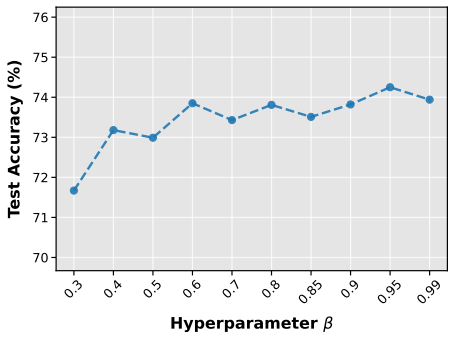
(c) CRDPLL, CIFAR-100



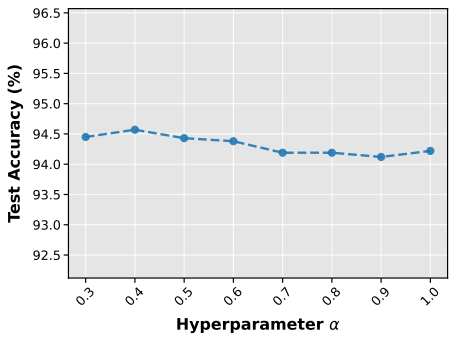
(d) CRDPLL, CUB-200



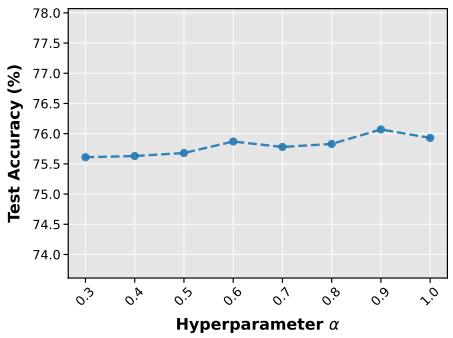
(e) β-Sensitivity (C-10)



(f) β-Sensitivity (C-100)



(g) α-Sensitivity (C-10)



(h) α-Sensitivity (C-100)

Figure 2: Convergence and hyper-parameter sensitivity analysis. (a)–(d) Convergence: test accuracy curves comparing the baseline (PiCO/CRDPLL) and our method (Ours) across different datasets. (e)–(h) Sensitivity: performance variations on PiCO CIFAR-10 (C-10) and CIFAR-100 (C-100) with respect to the momentum coefficient β (e–f) and the initial guidance weight α (g–h).